

# **DESIGN, IMPLEMENTATION, AND TEST OF A WIRELESS PEER-TO-PEER NETWORK FOR ROADWAY INCIDENT EXCHANGE**

Trevor Harmon, James Marca, Ray Klefstad, and Peter Martini

*Institute of Transportation Studies*

*University of California Irvine*

ATMS Testbed Technical Report TTR3-11

This work was performed as part of the ATMS Testbed Research and Development Program of the University of California, Irvine. The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views of polices of the State of California. This report does not constitute a standard, specification, or regulation.

**June 2005**

# DESIGN, IMPLEMENTATION, AND TEST OF A WIRELESS PEER-TO-PEER NETWORK FOR ROADWAY INCIDENT EXCHANGE

Trevor Harmon, James Marca, Ray Klefstad, and Peter Martini

Institute of Transportation Studies  
University of California, Irvine

## SUMMARY

The existing paradigms for vehicular traffic monitoring and control have a strong infrastructure bias—data is collected centrally, processed, and then redistributed to travelers and other clients in response to requests. There are a number of research efforts to decentralize traffic state monitoring by leveraging advanced local area wireless technology. Our version of such a traveler-centric system is called Autonet.

This report presents a preliminary *implementation* of some of the key Autonet concepts as well as some field measurements. An in-vehicle client with an informative GUI has been implemented. This client continuously listens for other clients, and exchanges knowledge about network incidents once contact is made. It is demonstrated that knowledge about traffic conditions can be propagated successfully using this system. The client programs were also used to test the actual throughput possible for messages sent from one vehicle to another using 802.11b wireless hardware. These measurements establish the maximum throughput at about 4,000 incidents for two vehicles moving in opposite directions at highway speeds.

## 1 INTRODUCTION

One of the key failings in the broad adoption of Intelligent Transportation Systems (ITS) technologies for providing traveler information to drivers is the lack of usable real-time data. According to a 2002 national survey of ITS technology adoption [ITS Joint Program Office, 2004], only about 30% of all signalized intersections on arterial streets had any form of electronic surveillance. That means 70% of all traffic lights have no electronic monitoring whatsoever, let alone the streets in between the signals. Thus even if every *highway* was fully and accurately monitored, drivers attempting to plot an alternate route around an incident would have essentially no information about conditions on the arterial street network.

During the same period covered by the ITS technology adoption surveys, automobile manufacturers have been cramming more and more computer electronics and control systems into their cars, with high end models commonly sporting in-dash, GPS-based mapping systems. The contrast between the high cost and low penetration rate of infrastructure based traffic monitoring, and the declining cost and increasing capabilities of in-vehicle electronics devices implies that we should explore the abilities of the latter to augment or replace the former. We suspect that a decentralized, vehicle-based traffic

probe system may be cheaper, faster, and easier to implement than the centralized monitoring and traffic probe systems defined in the National ITS Architecture [ITERIS, 2004].

We envision a decentralized, autonomous communications network between vehicles which leverages the unlicensed spectrum provided by the government and the off-the-shelf hardware that has been developed to exploit this spectrum (802.11a/b/g, DSRC ITS band etc.). The system we propose expects individuals to share and swap their own travel data freely, and allows each traveler to buy as much or as little a device as they need. We call this concept the Autonet, with the name deliberately resonating with the Internet, while still recognizing this network's autonomous and automotive (self-propelled) nature.

Past studies, including Ziliaskopoulos and Zhang [2003], Nadeem et al. [2004], Wu et al. [2004], Hartenstein et al. [2001], Schwingenschlögl and Kosch [2002], and Kher et al. [2002], have shown (primarily through simulation) that ad hoc wireless networks formed between moving vehicles are feasible for relaying traffic information. Our own analytical and simulation research has supported this view.

This report documents an initial, practical implementation of the core vehicle to vehicle client application. The primary goal of this work was to move from simulation to the real world. The results document the *current* capabilities of hardware for vehicle to vehicle communication of traveler information, provide working protocols and communications parameters for improving future large-scale simulation studies, and give us an in-vehicle client application which will be used in the future for studies of driver interaction and market acceptance.

The report is organized as follows. The next section presents the background for the work presented, describing how the larger Autonet idea relates to our practical implementation, as well as related work in this area. Section 3 then presents an initial design for a working Autonet prototype. Our primary application area—relaying changing traffic state information—played a significant role in limiting the grand vision to just its core functions to make implementation practical. Section 4 presents the implementation of our initial design into three working Autonet nodes. We present the results of a scenario test in Section 5, along with absolute throughput values for a variety of conditions. The results show that the general concept is sound, that vehicles can be expected to propagate incident information among themselves without reference to a centralized authority. Improving the basic algorithms and expanding the core functionality of an Autonet node towards a full scale Autonet are the subject of continuing research efforts.

## **2 BACKGROUND**

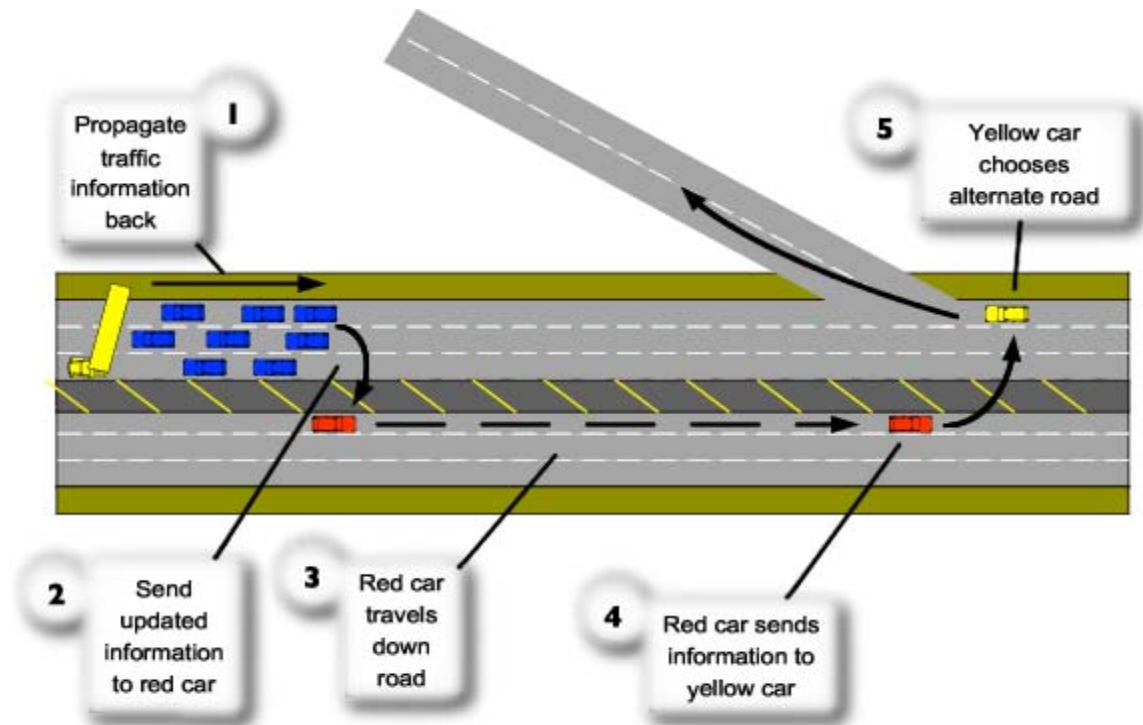
At its core, the Autonet concept is that electronic devices communicate with each other wirelessly in order to provide travelers with useful information. This core concept can be broken down into component questions.

1. What information is useful to a traveler that other vehicles know?
2. How can that information be presented?
3. What wireless technologies are suitable for sharing that information between vehicles?

4. What kind of an in-vehicle device can accomplish all the necessary tasks?

### 2.1 Useful Information

First, we need to determine what information is useful to the decisions that a traveler needs to make. For the initial Autonet application, we deliberately attempted to keep things as simple as possible. Specifically, we focused on the core task of driving—getting from one place to another in the minimum amount of time. Conventional travel behavior theory places a large cost on travel time, and supposes that travelers attempt to minimize their travel time. Thus at a minimum, the in-vehicle device acting independently of all other devices should be able to compute the shortest path between two points given expected travel time on each link.



**Figure 1. By running variations of the basic scenario shown here on actual roadways, we could judge the effectiveness of our prototype and ensure that our algorithms were performing as expected.**

Further, it is known that people make their choices based on imperfect information. They do not know the exact conditions on all links in the network, and the expected travel time can be wildly incorrect given current conditions. Therefore, one important goal of the Autonet is to enable vehicle to vehicle exchange of information related to conditions that the traveler does not know. This information might range from revised road geometries due to work zones, to historical traffic conditions on a link, to real-time incident alerts.

Vehicles must identify changes from baseline conditions, and communicate these changes to other vehicles. In our view, this is the crux of the Autonet system. Vehicles must be able to identify situations which are unexpected, without reference to a central authority. They need to be able to transmit this information to other vehicles. And there

must be a mechanism in place to arbitrate between the information gathered by all vehicles so that a correct interpretation of current conditions can be established.

We explored various ideas for communicating information and decided on one basic scenario that would be extremely useful for propagating information. Shown in Figure 1, this scenario involves passing a message about traffic state from one direction of traffic flow to the other for rebroadcast to upstream vehicles. While the figure shows the upstream vehicle choosing to exit the freeway, it is more likely that the new knowledge about downstream conditions would be added as evidence to an inference engine to anticipate the most likely conditions in the near future, and to compute the optimal path accordingly.

## ***2.2 Presentation of Information***

Moving to the next question, the data collected by an Autonet device needs to be presented, both to the traveler and to peer Autonet devices. We have designed our prototype system with a graphical interface designed to be used by a passenger (a navigator). While a driver interface is best left to UI experts, designing an informative graphical display is an excellent way to evaluate what information can be presented economically, as well as to anticipate what kinds of queries or other user input can be expected. More discussion of the user interface can be found in Section 4.

Presenting novel information to other vehicles is perhaps even more important than interfacing with the driver. Details on the initial design of the communication protocol are given in Section 3.1. The purpose of this project was to start the iteration of design and test by getting an initial prototype on the road to evaluate the limitations of today's commercial off-the-shelf (COTS) technology. We have chosen to build a hardware-in-the-loop experiment at this early stage specifically to determine what is possible, and what tasks the Autonet can reasonably be expected to perform. A field experiment avoids possible mistakes in evaluating the capability of current hardware, the real limitations and compromises required for user interface design, and errors due to assumptions embedded in the simulated model. This last point is especially important for wireless hardware, as line-of-sight is crucial for good communication and is nearly impossible to capture accurately in a simulation.

## ***2.3 Suitable Wireless Technology***

For device to device communication, the easiest solution is to use a local area wireless link, rather than a point-to-point wide area link such as a 3G cellular. This is not at all a closed issue, however. The long-standing concept of a probe-vehicle states that the collected traffic flow data from a vehicle should be uploaded directly to a central computer for processing. Indeed, our own GPS-based travel survey devices do just that over a CDPD modem.

Central to the question of choosing a suitable wireless technology is our understanding that the vast majority of this “novel” information required by Autonet participants has little or no value to the traveler who actually observes the information in question. Specifically, if a traveler has just passed over a section of road, has just measured the

travel time across that link, and has entered that bit of data into their network conditions inference engine, then that information has no further value to the traveler. But the Autonet concept requires that the traveler keep that bit of information around, and then broadcast it to other vehicles.

Because data storage is relatively plentiful, keeping this information should have no added cost. But the same cannot be said about the act of transmitting that data out to others. First, the use of the wireless link must have a zero marginal cost to the driver. This condition is satisfied with license-free technologies such as 802.11b. This condition might also be satisfied by 3G cellular type services if users pay a flat monthly fee for unlimited data usage. We will revisit this question in the future, but in terms of today's COTS technology, 3G cellular does not exist everywhere, whereas 802.11b does.

In addition to the cost of using the wireless link, there is also an appreciable opportunity cost to broadcasting data, given the limited bandwidth that is available. Sending one bit might mean that the driver cannot hear another bit, perhaps one that has a great deal of value. So it might be in the driver's best interests to listen rather than broadcast. We may attempt to address some of these issues by tracking the speed of the vehicle and throttling back broadcasts at high speeds, but this is clearly an area which will require much more research in the future as the system matures.

Along with the decision to use 802.11b hardware came a host of other questions. What we found appeared to be somewhat conflicting. Ebner et al. [2003] recommend against using 802.11b, stating that their simulation results indicated that its performance would be unacceptable in urban environments and at high speeds. In contrast, Singh et al. [2002] conducted several field tests and concluded that 802.11b performance is "suitable for inter-vehicle communications." These papers are not in conflict, but rather bind up the concept of "suitability" with the two different applications that the authors have in mind. Ebner et al won't get a seamless approximation of the Internet on the road over 802.11b, while Singh et al. have a much more modest goal of vehicle to vehicle communications. Nadeem et al. [2004] have designed their system around 802.11b, and report some results mixing real world tests and simulation for vehicles moving in traffic stream. In Wu et al. [2004], a detailed protocol for sending a message from one geographic area to another is developed, and is shown to work in simulation using 2 Mbps 802.11 DCF MAC protocol. Finally, note that 802.11b has been used in infrastructure mode to accomplish vehicle to vehicle communications (communication between two vehicles with fixed IP addresses while traveling along a corridor) in Aziz [2003].

Many other papers exist which address the characteristics of 802.11b and its relatives (including the new Dedicated Short Range Communications (DSRC) Service for Intelligent Transportation Systems (ITS) in the 5.9 GHz band). The suitability of 802.11b (or a, g, or DSRC) for vehicle area, ad hoc networking comes down to the requirements of the application. If one desires to achieve "Internet on the road" as in Ebner et al. [2003], then the network must allow a roadside node to address any particular mobile node possibly over several hops. This in turn requires maintaining a dynamic routing table, or developing algorithms to establish routes on demand. If, on the other hand, one wishes to automatically form a high-speed platoon, as with automated highway systems (AHS) research, then the wireless hardware does not need to allow multihop networking, but must have extremely low error rates, as dropped packets could be fatal.

For the Autonet, vehicle to vehicle communication is intended only for informational purposes, not for vehicle control. A communication failure will result in a few bits of missed information. While this may lead to suboptimal decisions (a wrong turn or a wrong route), communications errors will not cause an accident. At the other end of the spectrum, there is no need to establish a complete, multi-hop network to obtain quantitative improvements in the level of system awareness for participating drivers. More connectivity would be better, but even a single hop network could be useful. It is expected that Autonet-equipped vehicles will be widely spaced, at least until adoption rates are extremely high. Given these modest requirements for the initial Autonet vision, we decided that 802.11b was more than adequate for building our field test units.

## **2.4 In-vehicle Device**

We accept as fact the proposition that within a year or two a cutting edge device will exist that can do all we ask of it and more. However, we are far more interested in what today's hardware can do, which will give an excellent indication of the capabilities of tomorrow's commodity hardware.

At a minimum, the Autonet-compliant device must have a global positioning system (GPS) antenna, a wireless link, a CPU, and sufficient storage capacity to process a reasonably sized travel network. Our test platform is a laptop, connected to a Garmin OEM GPS antenna. We have also ported the system to a custom embedded system platform based on the PC/104 form factor.

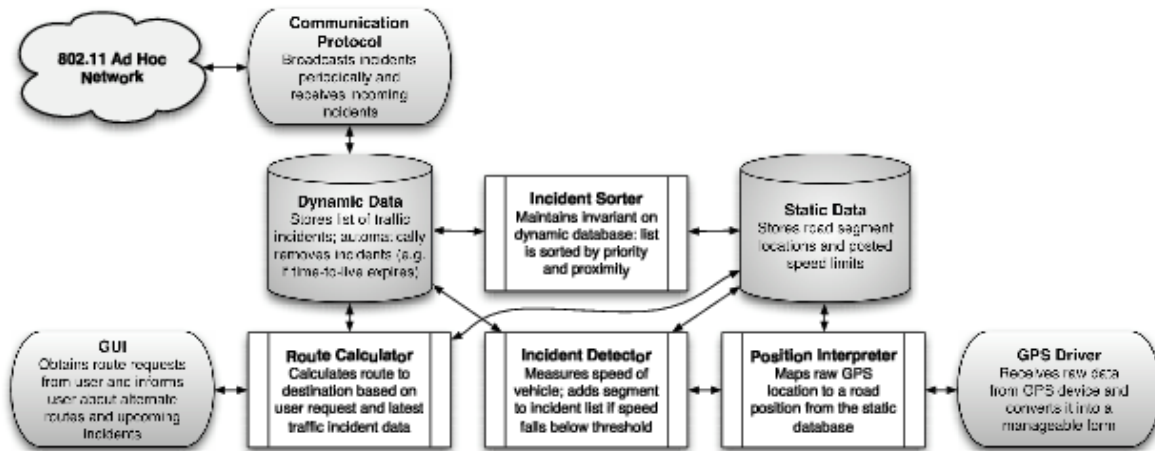
It is also interesting to note that cellular telephones are beginning to appear with GPS antennas built into them. The only problem with readily available cell phones is their lack of local area wireless such as 802.11b. At the time of writing, this is beginning to change, as telecom companies with land-line resources such as British Telecom attempt to lure customers with dual mode voice over IP (VOIP) and cellular telephones (BT's announced BT Fusion demonstration, for example), and as cellular carriers respond to the ability to make completely free calls using Skype and a PDA or laptop at a WiFi hotspot. Some of the high end cell phones that the *manufacturers* have announced on paper could run our Java-based application, but it is difficult to purchase these phones from cellular carriers. Still, we are confident that in five years consumers will own Autonet-compliant cell phones. At that time, participation in the Autonet would mean simply running Autonet client and server code on those phones while they are driving.

## **3 DESIGN**

As a first step in reaching the goals of Autonet described in Section 2, we have developed a prototype Autonet computer. This prototype demonstrates the feasibility of Autonet and provides a platform for testing our algorithms in a realistic environment on public roads. It also serves as a testbed for examining commercial off-the-shelf hardware and whether the benefits of such hardware (namely, lower cost due to economies of scale) can be applied to Autonet. In Section 5, for example, we discuss how we used our prototype to test the ability of 802.11-based wireless data technologies to send traffic information between vehicles at highway speeds. In the paragraphs that follow, we describe the fundamental components of our prototype and provide some insight into its design.

### 3.1 Software Overview

The heart of the Autonet system, then, is the software. Figure 2 shows a diagram of our prototype's major software components and the relationships between them. We have divided them into distinct components for modularity: We expect that future Autonet prototypes will provide vastly improved implementations over our initial version, and by separating code into modules as shown in the figure, we can easily plug in new implementations as necessary.



**Figure 2. The major software components of the Autonet prototype. Cylinders represent data; rectangles represent sub-processes; squashed rectangles represent software glue between sub-processes and a hardware device.**

Each of the components is designed to operate concurrently with the others, performing operations on incidents as they come in and responding to user requests asynchronously. For clarity, however, the components can be thought of as operating in a sequential fashion as in the following scenario:

1. The Communication Protocol receives a message from a passing Autonet vehicle. The message contains information about an incident: location, time observed, and severity (e.g., deviation of observed speed from expected speed).
2. The Dynamic Data component receives the new incident and checks it for validity. If, for example, the incident is out of date, it will not be placed into dynamic storage.
3. The Incident Sorter observes the change in dynamic data and re-sorts it according to some evaluation function. This function takes into account the proximity of the incident, the elapsed time since it was first observed, and its severity, among other criteria. (We define one such function for our initial prototype, as discussed in Section 3.2.)
4. The Route Calculator also observes the change and recalculates the shortest route to the user's desired destination. It takes into account the location of the new incident and re-routes the path to avoid it, if possible.
5. Finally, the Communication Protocol queries the database for the highest priority incidents, as this set may have changed with the addition of new



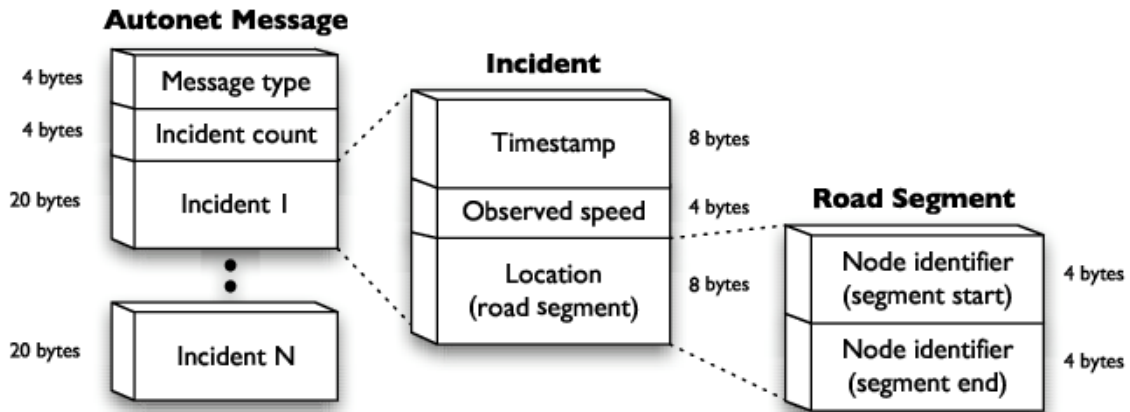
incidents. It begins broadcasting the set so that other Autonet-enabled vehicles can know about the new incident.

Some of the sub-processes shown in Figure 2 do not appear in this scenario because they are largely independent. The Position Interpreter, for example, runs periodically to adjust the vehicle's current position (obtained from the GPS Driver) to correct for errors and make sure that the displayed position does not move off of the road. Similarly, the Incident Detector is independent and asynchronous. It is highly implementation-dependent, as well, because a variety of techniques may be used to detect incidents, and the very definition of a traffic incident, especially as it relates to Autonet, is still a subject of research. For these reasons, we do not discuss the Incident Detector here; instead, we describe in Section 4.1.2 a simple incident detection technique that we employed for our Autonet prototype.

Of course, all of the software components from Figure 2 may vary substantially in any given implementation, depending on hardware constraints, the latest research findings, and the particular goals of the software developer. However, we have settled on a design for two components in particular—the communication protocol and the data structures—that we believe are appropriate for any Autonet implementation, and we describe them in detail in the following sections.

### 3.2 Communication Protocol

The communication protocol is perhaps the most important piece of Autonet's software, as it has the greatest impact on the overall quality of the system. It is responsible for exchanging incidents with other Autonet-enabled vehicles, and it ensures that its user obtains the maximum possible knowledge of the traffic state. It accomplishes these feats by broadcasting a list of incidents in the format shown in Figure 3. It also listens for incoming messages of the same format.



**Figure 3. Autonet vehicles exchange traffic information with each other by broadcasting a list of incidents in the format shown here.**

In the figure, “message type” refers to a pre-defined value that indicates the type of the message. Currently, we define only one type, BROADCAST, but we include a type value

in our message for future expansion. For instance, we may want to add an additional type, REQUEST, so that one vehicle may query another for a specific piece of information. (See Section 5 for a discussion of the feasibility of this idea and why we chose not to implement it in our initial prototype.)

The next data element is a positive integer declaring how many incidents are being sent with the message. This number is followed by the incidents, each of which contains a 64-bit timestamp that indicates the number of milliseconds that have elapsed since midnight (UTC time) on January 1, 1970 and the time when the incident was first observed. Each incident value also includes an integer representing the speed, in kilometers per hour, that was observed at the incident. Finally, the incident structure provides the location of the road segment, indicated by the 32-bit globally unique identifiers of its starting and ending points, where the incident occurred.

The message format for incident exchange is simple; the decision of which incidents to send is not. The problem is that over time, a group of Autonet-enabled vehicles may build up a list of tens of thousands of incidents. According to our measurements (see Section 5), two vehicles passing each other in opposite directions can exchange only a few thousand incidents each when using 802.11 technology, even under the best of conditions. Therefore, the communication protocol must decide which portion of the incident list to broadcast.

In our prototype, we address this problem by assigning a priority to each incident, calculated with an evaluation function that takes into account the following parameters:

- **Deviation from expected speed** The higher the delta from the expected speed, the more important is the incident. Currently, we define expected speed as the posted speed limit for the road, but for future prototypes we will define it more accurately as the historical average of vehicles passing on the road segment.
- **Size of the road** Highways have higher priority than small country roads. Distance from the local position Traffic information from another state, for example, is of lower importance than information regarding a nearby road.
- **Time** The older the information, the less accurate it is, and therefore the less important.

The Incident Sorter applies this evaluation function to sort the incident data by priority. The communication protocol then skims off the highest priority incidents and broadcasts them continuously so that passing vehicles can collect them. Note that this technique does not fully solve the problem, however.

Specifically, there is the question of how many high-priority incidents to broadcast. If the communication protocol broadcasts too many, passing vehicles may lose the highest priority incidents because the window may be too narrow, and their communication overlap might occur at the tail end of the broadcast list. Likewise, if the Incident Sorter selects too few incidents to broadcast, then passing vehicles with large communication overlap windows will receive duplicate incidents, and the bandwidth of the channel goes to waste.

To counter this problem, we have modified the communication protocol to adjust the number of incidents that it broadcasts according to the speed of the vehicle's travel.

When moving at highway speeds, for example, it broadcasts only 1000 incidents, which our measurements show is an amount that can be transferred easily at such speeds (see Section 5). As the vehicle slows, the communication protocol increases the amount of incidents it cycles through during a broadcast period, transferring up to 5000 incidents in a single cycle. By adapting to changing conditions in this manner, our communication protocol takes full advantage of the window of opportunity in which two passing vehicles are within signal range.

Although this technique addresses the problem of high-speed incident transfers, a separate problem arises at slow speeds. If, for instance, a large group of Autonet-enabled vehicles stops at an intersection, each will flood the channel with incidents, possibly causing collisions that drastically lower effective bandwidth. We mitigate this problem by inserting random pauses between broadcast cycles, freeing the channel momentarily for other vehicles. Similar in concept to bus arbitration in Ethernet, this technique allows Autonet to take full advantage of long waits at intersections by using them for incident transfer among vehicles.

Finally, any evaluation function for incident priority is imperfect. It cannot predict with 100% accuracy the value of an incident to surrounding vehicles. For instance, an incident may seem unimportant to a vehicle that has already passed it by, but it has high value for vehicles approaching from the opposite direction. For this reason, we have altered the communication protocol such that a fraction of the incidents for broadcast are based not on the evaluation function but instead drawn randomly from the set of all known incidents. We believe that only a small fraction of this randomness—10%—is enough to counter any inaccuracies in our evaluation function, although we cannot say this conclusively without additional research, as we discuss in Section

### ***3.3 Data Structures***

In Autonet, data can be broken down into two basic types: static and dynamic.

#### Static data

Static data is essentially a map of the road system in which an Autonet-enabled vehicle may travel. We call it “static” data because the road system changes relatively infrequently, requiring far fewer updates compared to traffic incident data.

At a minimum, static data consists of a graph of road segments, where each road segment is an edge of the graph, and each edge is bound to a globally unique number for identification. Although not strictly required for the basic operation of Autonet, our design includes additional information for each segment, such as the road name (for a more pleasant user interface) and the posted speed limit (for incident detection, as explained in Section 4.1.2).

Together, all of this information requires approximately 100 kilobytes for a typical college campus or 13 megabytes for a highly populous county (2,500 square kilometers, 85% land, 1,400 people per square kilometer). Using current off-the-shelf hardware, one can reasonably expect an embedded device to have anywhere between 64M and 128M of RAM, which can handle these smaller areas. The large amount of storage required for,

say, North America would easily overflow the memory capacity of a reasonably-priced Autonet computer using today's technology. Thus, any practical, location-independent Autonet system would require secondary mass storage such as a DVD-ROM disc, or access to such a resource over a high-speed wide-area wireless link. This data repository would be accessed and swapped in and out of RAM as needed.

### Dynamic data

Dynamic data consists of traffic incidents: portions of the road where vehicles are moving slower than expected. Unlike static data, this information must be placed entirely into RAM because it is constantly read and written by the communication protocol in order to broadcast existing incidents and collect new ones from passing vehicles. The dynamic data is also much smaller in size than the static data because only deviations from normal traffic flow must be stored.

Optimizations can further reduce the size required for storing incidents. For example, the static database requires multiple segments to represent a curve in the road. If a traffic jam occurs on this road, multiple vehicles may report incidents on each segment, even though only one incident is necessary to represent the traffic jam, assuming there are no intersections on the curve. The code for managing the dynamic data can therefore reduce several redundant incident reports down to one, resulting in substantial space savings.

Space can also be saved in the dynamic database by eliminating incidents that have expired. More of a requirement than an optimization, this feature ensures that stale incidents—that is, incidents that have exceeded their time-to-live tag and thus may no longer be valid—do not adversely affect routing decisions made by Autonet on behalf of the user. Expiring incidents in this manner can also reduce congestion within the Autonet network.

## **4 IMPLEMENTATION**

The goals were compelling, and our design appeared sound, but to show convincingly that the very idea of Autonet was possible, we needed to construct actual working prototypes. We therefore decided to build three Autonet devices that we could install in vehicles and test on public roads. Three was deemed the minimum necessary to prove the viability of Autonet because it would enable us to run experiments that were representative of typical Autonet tasks. For instance, we wanted to show that one vehicle could propagate an incident to another, even if they are out of signal range, by enlisting the help of a third vehicle. We also wanted to show that this scenario was possible in a harsh operating environment (not just in simulation) using commercial off-the-shelf hardware.

We began our implementation with three basic laptop computers. In each laptop, we installed an 802.11b wireless Ethernet card and configured it for “ad hoc” mode, allowing network connections to be established between the laptops automatically, as soon as they come within range of each other.

With the computer and wireless network device in place, the only hardware we still required was a GPS receiver. Initially, we used receivers from Garmin that we connected

to the laptops via serial cable, and we relied on a daemon called `gpsd` [Raymond et al., 2004] to parse the position and velocity data sent by the GPS device. Later, we switched to receivers from TeleType that send GPS data over Bluetooth connections to the laptops. Not only was this configuration more convenient due to the lack of wires, it also allowed us to test the performance of Autonet under adverse conditions, since Bluetooth operates at the same frequency (2.4 gigahertz) as the 802.11b devices we were using for Autonet’s communication protocol. As our measurements in Section 5 show, the interference between the two devices was negligible.

#### ***4.1 Software Components***

For software, we installed Sun’s Java runtime on each machine. Java was our language of choice for implementing Autonet due to its ease of use and solid support for network communication and graphical user interfaces, both of which we desired for our implementation. We then proceeded with the grunt work of implementing the various software components from our design description in Section 3. We highlight three of these components—Static Data, the Incident Detector, and the GUI—in the following sections.

##### Map data

For static (map) data, we were faced with essentially two choices: purchase map data from a commercial vendor such as NavTech, or create our own from a freely available source such as TIGER/Line. We found that commercial data was prohibitively expensive, however, so we chose the latter option. We developed software that parses TIGER/Line files, a digital road database provided to the public by the U.S. Census Bureau.

Parsing is a two-step process: one step to extract the features for a particular region of interest, and another to convert the data to our custom format that, in order to save space, contains just enough information for incident tracking and route calculations.

Unfortunately, the results were not pleasing. The problem lay not in our code but in the quality of the source data. For example, we noticed that a number of roads, even major highways, contained missing segments. Furthermore, TIGER/Line data does not provide information about one-way roads, an important criterion for proper route calculation. We are aware of on-going efforts by the US Census to improve the TIGER/Line files, and so these problems may just go away. Furthermore, it is reasonable to expect that widespread adoption of route guidance applications that use the TIGER/Line data as a primary source will in turn spur improvements in that source database. Thus while we cannot claim that a real-world routing application is currently possible using TIGER/Line data, we would encourage its use as a primary source in the hopes that such use will improve the quality of this excellent, free resource.

##### Incident detection

Our Incident Detector, as discussed in Section 3.1, provides a very basic algorithm for detecting new traffic incidents. By using speed limit information stored in the static database, together with the current speed obtained from the Position Interpreter, it

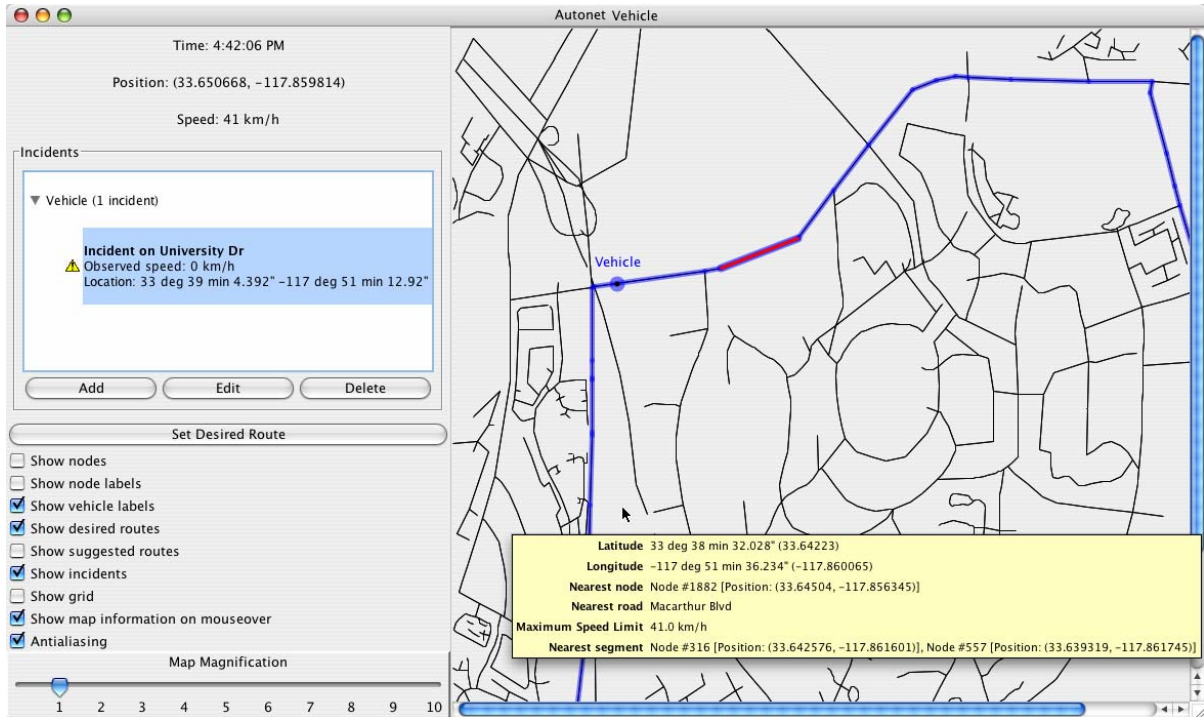
declares an incident as soon as the vehicle drops below 80% of its expected speed. Although this technique works well for highway driving, it is inadequate for other roads, where a stoplight can inject false incidents into the system.

Although we have a number of ideas for mitigating these problems, much work has already been done in the transportation research community on incident detection [Cheu et al., 2002; Taylor et al., 2000; Thomas, 1998; Sermons and Koppelman, 1996; and Sethi et al., 1995], and we did not wish to re-invent the wheel for our initial prototype. As we settle on a known set of capabilities and resources for an Autonet peer, we will turn to integrating the rich body of literature on incident detection techniques into the unique requirements of the Autonet devices.

### User interface

In its final form, Autonet will provide a graphical interface for the user, perhaps by means of a display embedded into the automobile's dashboard. This display would show the current state of the automobile—speed, heading, and so on—and allow the user to indicate a desired route. To the casual viewer, this display would appear no different from commercial navigation systems that have become popular in luxury automobiles. The Autonet user interface differs from these more conventional systems, however, by displaying up-to-the-minute traffic information collected from other Autonet-enabled vehicles. By combining this information with the driver's chosen route, Autonet's interface would then be able to display a more appropriate route (if found) superimposed over a map of the vehicle's surrounding area.

Strictly speaking, the design and implementation of a graphical user interface for Autonet is not necessary until it is ready to be deployed to the general public. Instead of waiting for this moment to arrive, we chose a more proactive approach: We developed a prototype user interface in parallel with our development of Autonet's algorithms, communication protocol, and other sub-systems. This approach proved to be quite valuable in guiding our work. For example, we were able to see first-hand the performance of our graph traversal algorithms simply by launching our user interface prototype and playing the role of the end-user. This rough but effective profiling technique helped us to judge and improve our code optimizations more quickly. It was also an excellent tool for debugging. The road map display, for instance, which we initially constructed only for the user interface prototype, revealed flaws in our source data (such as broken road segments) that would otherwise have gone undetected.



**Figure 4. By constructing a prototype Autonet client user interface, as shown in this screenshot, we could easily try out new feature ideas and identify performance problems in our code.**

Figure 4 shows the results of this effort. The left-hand column of the screen shows the vehicle’s speed and position as reported by the GPS device, followed by a list of incidents known by the vehicle. (This list includes second-hand information received from Autonet-enabled partners as well as traffic incidents observed directly.) The bottom half of the column provides controls for adding, editing, and removing incidents and for manipulating various attributes of the display, such as map magnification. The right-hand side of the screen shows a road map of the vehicle’s surrounding area, along with its position on the map and its chosen route. Note that in this screenshot the map also shows the location of an incident detected on University Drive.

## 4.2 Mission Control

Although the user interface described above is a valuable aid for testing our Autonet implementation, it is not very convenient. During field tests, each vehicle could observe its local state, but there was no way of finding the global state, useful for debugging and analyzing the performance of Autonet as a whole. We attempted to circumvent the problem by communicating via cell phones, but the awkwardness of this technique quickly made us realize that a better solution was necessary.

This desire for global knowledge of Autonet’s state during our field tests was the impetus for “Mission Control,” a desktop application we created exclusively for testing and debugging Autonet. It closely resembles the user interface shown in Figure 4, but instead of displaying one vehicle’s state, it shows all of them simultaneously.

To accomplish this, we equipped each Autonet prototype with a GPRS modem that periodically sends its vehicle's state—position, speed, incident list, and desired route—back to Mission Control, which collates all of the incoming data and displays the vehicles' movements in real time. Mission Control can also send messages back to the remote vehicles that command them to add artificial incidents or alter desired routes, allowing a single operator to control all of the Autonet vehicles from a central location.

With the help of this tool, we could execute an Autonet scenario out in the field while one team member remains back at the lab ("Houston") to observe the interactions and changing states of the remote vehicles. Thus, we could set up experiments and pinpoint problems much more easily, avoiding the constant "What is on your screen now?" dialogue that plagued our earlier phone-based approach to debugging.

## **5 EMPIRICAL RESULTS**

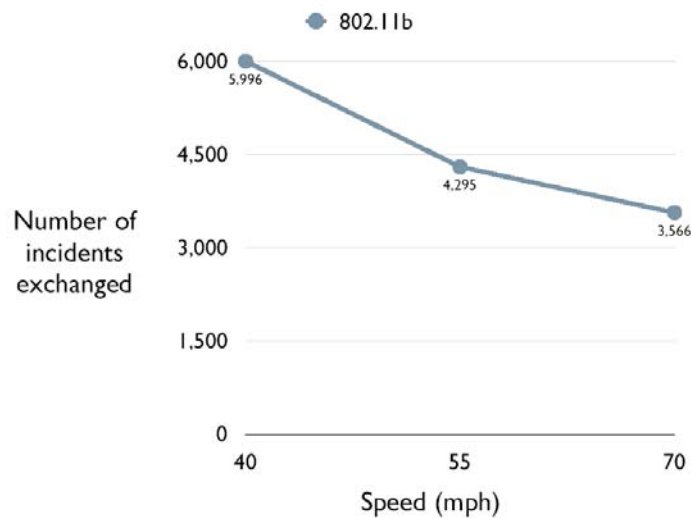
Rather than simple benchmark testing of this prototype, we focused instead on a real-life scenario to expose some of the more tangible benefits of Autonet's design. In this scenario, shown earlier in Figure 1, three vehicles (at a minimum) equipped with identical versions of our Autonet prototype move along a highway, one in a direction opposite to the others.

One vehicle, shown on the right-hand side of the figure, is stopped due to an accident on the highway, and it begins broadcasting the incident to surrounding vehicles. A second vehicle, traveling in the opposite direction, receives the broadcasted incident and proceeds to broadcast it, as well. After some distance, it passes a third Autonet-enabled vehicle that receives the original incident, despite being out of the first vehicle's range, by receiving transmissions from the second vehicle. This third vehicle can then use the new information to adjust its route and avoid the traffic jam blocking the road ahead.

We executed a number of variations of this scenario using our prototype on nearby highways and streets. After several revisions of our software, we were able to perform the scenario successfully, demonstrating conclusively that our Autonet prototype worked as expected. Thus, we showed that Autonet can be implemented using conventional off-the-shelf technology and perform well under harsh, realistic conditions.

In addition, we ran a variety of benchmarking tests that we had created for testing the performance of 802.11b technology as it applies to Autonet. Figures 5, 6, and 7 show the result of this effort. We collected all of these measurements with a Lucent Technologies ORiNOCO 802.11b network adapter connected to a 5.5 dBi external antenna.



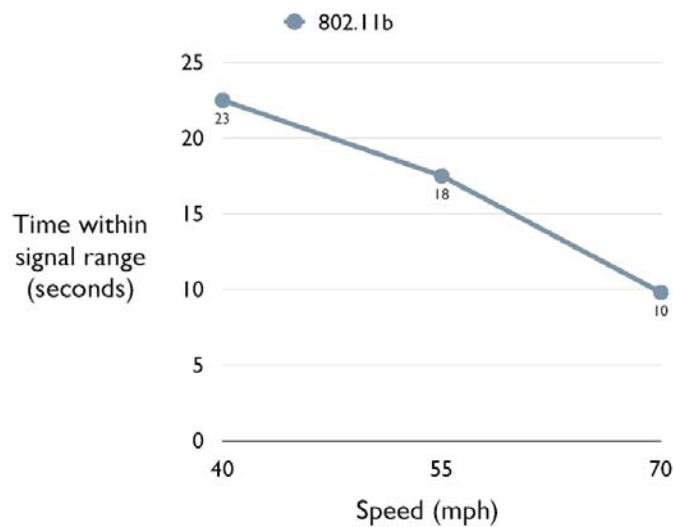


**Figure 5. This graph shows the results of our 802.11b throughput benchmark, where two cars exchange Autonet incidents as they move in opposite directions.**

For Figure 5, we tested the ability of 802.11b to transfer incidents between two Autonet-enabled vehicles that pass each other in opposite directions at identical speeds. The horizontal axis refers to the speed of each vehicle (not relative speed), and the vertical axis indicates the average number of incidents that each vehicle could send. At 40 mph, for example, a total of nearly 12,000 incidents on average was sent and received on each single pass.

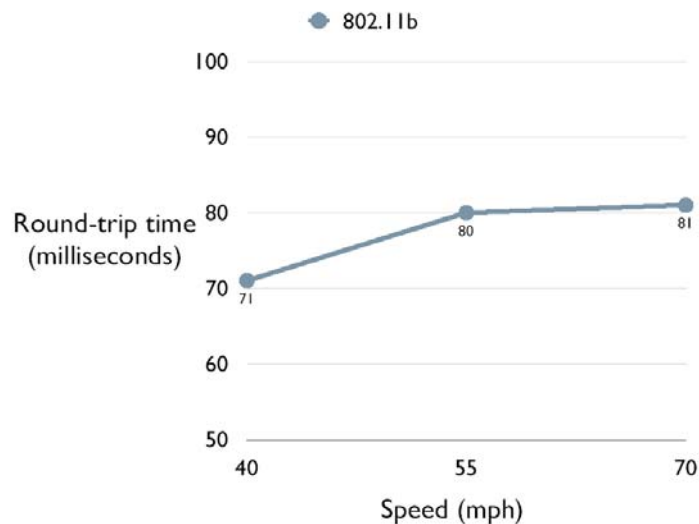
As expected, the number of incidents exchanged drops sharply as the speed of the vehicles increases. The decline is largely due to the shorter time in which the two vehicles are within signal range, but we suspect that the Doppler effect also plays a role in limiting throughput with increasing speed. We note, however, that even when two cars travel at highway speed--70 miles per hour—they are able to send and receive over 3,000 incidents each. These are very promising results, as they show that one vehicle can inform another about every single segment of road for a typical urban area of 64 square kilometers. If we assume that one out of every five segments is an incident, the area grows to 324 square kilometers.

For Figure 6, we measured the amount of time during which two vehicles, when traveling in opposite directions, could remain in 802.11b signal range of each other and successfully exchange incidents. Recall that we collected these measurements with the help of an external antenna; this explains the surprisingly long periods during which the two vehicles were able to remain in contact. Although we had hoped to collect data without this antenna for comparison purposes, we were unable to exchange incidents reliably after removing it, even at slow speeds. The consequence for Autonet is that any production version of our prototype would require the consumer to install an external antenna in his or her vehicle.



**Figure 6. This graph shows the amount of time two vehicles, passing at identical speed in opposite directions, can stay within signal range of each other using 802.11b technology and external antennas.**

Finally, Figure 7 shows the round-trip time for message exchange using 802.11b. Here, we define “round-trip time” as the time required for one vehicle to send an incident request to another and fully receive a response to the query. Based on these measurements, combined with the measurements from Figure 6, we calculate that when using a request/response method of incident exchange, approximately 200 incidents can be received per vehicle.



**Figure5. This graph shows the round-trip time for message passing between two vehicles traveling at the same speed in opposite directions.**

This number is an order of magnitude lower than the number of incidents a vehicle is able to receive when using the broadcast method of incident exchange. This is one reason why we have focused on the broadcast method of exchange for our Autonet prototype. On the other hand, the number of messages that *need* to be passed between any two vehicles is a function of the number of Autonet-compliant vehicles, and the level of system awareness that is desired and possible. If a vehicle has one thousand unique incidents, but 90% of them are old, then 100 messages is enough. Alternately, if two platoons of ten cars each are passing each other, and if each car is responsible for disseminating some fragment of the platoon's collective knowledge, then 100 messages per car is also probably enough. Finally, there is also the possibility to develop a hybrid messaging protocol, in which the usual request/ acknowledgment messages are followed by a broadcast period. This hybrid method may prove to be the best solution. These questions need to be explored with further research and development.

## **6 FUTURE WORK**

The prototype version of an Autonet computer has successfully demonstrated that the basic Autonet idea is sound—that vehicles can be expected to share information about traffic conditions in a real-world environment. In addition, the exercise of turning theory into reality has shown several areas which need improvement. Chief among these are deficiencies in the way novel information is detected and shared with the driver and with other Autonet peers. For the former, we are actively researching Bayesian inference techniques as in Liao et al. [2004], which will help determine what information is relevant to the current trip. Improving and expanding the Autonet middleware is another area of active development, which will allow decentralized group coordination of event detection, measurement, and description. In order to address questions related to scalability as penetration rates increase, we will be integrating the Autonet node capabilities into the Paramics simulator as a plug-in. Finally, there is no requirement that Autonet peers be vehicles. Specifically, we will be developing prototype devices for pedestrians and for roadside nodes, with the latter allowing traffic authorities to tap into the knowledge available in the Autonet to improve signal timing and coordination.

## **7 CONCLUSION**

The primary goal of the work documented in this paper was to prove that the Autonet concept would work in the real world using COTS hardware. The results show that the basic incident handoff scenario can be achieved. Detailed tests of the communications software developed over 802.11b show that at least a thousand incidents can be exchanged between vehicles moving in opposite directions at 70 MPH (relative speed 140 MPH). This implementation provides working protocols and communications parameters for improving future large-scale simulation studies. Finally, the prototype allows us to conduct attitudinal studies of drivers to explore questions about adoption rates, privacy concerns, and overall market acceptance.

## REFERENCES

- Farhan Muhammad Aziz. Implementation and analysis of wireless local area networks for high-mobility telematics. Master's thesis, Virginia Polytechnic Institute and State University, 2003.
- Ruey Long Cheu, Chi Xie, and Der-Horng Lee. Probe vehicle population and sample size for arterial speed estimation. *Computer-Aided Civil & Infrastructure Engineering*, 17(1):53-60, January 2002.
- A. Ebner, H. Rohling, L. Wischhof, R. Halfmann, and M. Lott. Performance of UTRA TDD Ad-Hoc and IEEE 802.11b in vehicular environments. In *IEEE Vehicular Technology Conference*, Spring 2003, Volume 2, pages 960-964, Jeju, South Korea, 2003. IEEE.
- Hannes Hartenstein, Bernd Bochow, André Ebner, Mathias Lott, Markus Radimirsch, and Dieter Vollmer. Position-aware ad hoc wireless networks for inter-vehicle communications: the Fleetnet project. In *Proceedings of the 2001 ACM International Symposium on Mobile ad hoc networking and computing*, pages 259-262. ACM Press, 2001. ISBN 1-58113-428-2.
- ITERIS. National ITS Architecture Version 5.0. website, U.S. Department of Transportation Intelligent Transportation Systems, 2004. URL {<http://www.iteris.com/itsarch/index.htm>}.
- ITS Joint Program Office. Tracking the deployment of the integrated metropolitan intelligent transportation systems infrastructure in the USA: FY2002 results. Technical Report Contract No. DTFH61-00-Y-30014, Federal Highway Administration, April 2004. URL {<http://itsdeployment2.ed.ornl.gov/its2002/downloaddocuments.asp>}. Prepared by Oak Ridge National Laboratory and Science Applications International Corporation.
- Shubhalaxmi Kher, Sanjiv Tokekar, and P.K. Chande. Self sustaining traffic management system and its compartmental modeling. In *Proceedings. The IEEE 5<sup>th</sup> International Conference on Intelligent Transportation Systems*, pages 639-644. IEEE, 2002.
- L. Liao, D. Fox, and H. Kautz. Learning and inferring transportation routines. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-04)*, pages 348-353, Menlo Park, CA, 2004. AAAI Press. URL [http://www.cs.washington.edu/ai/Mobile\\_Robotics/projects/postscripts/gps-aaai-04.pdf](http://www.cs.washington.edu/ai/Mobile_Robotics/projects/postscripts/gps-aaai-04.pdf). Made available by the author on the Internet.
- Tamer Nadeem, Sasan Dashtinezhad, Chunyuan Liao, and Liviu Iftode. TrafficView: Traffic data dissemination using car-to-car communication. *Mobile Computing and Communications Review*, 8(3):6-19, July 2004.
- Eric S. Raymond, Remco Treffkorn, Derrick Brashear, and Russ Nelson. A GPS service daemon, 2004. URL <http://gpsd.berlios.de/>.
- Christian Schwingenschlögl and Timo Kosch. Geocast enhancements of AODV for vehicular networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(3):96-97, July 2002.

- M. William Sermons and Frank S. Koppelman. Use of vehicle positioning data for arterial incident detection. *Transportation Research Part C: Emerging Technologies*, 4(2):87-96, April 1996. URL <http://www.sciencedirect.com/science/article/B6VGJ-3Y2N033-3/2/290ec1a0cae693bca07692470d83bb2a>.
- Vaneet Sethi, Nikhil Bhandari, Frank S. Koppelman, and Joseph L. Schofer. Arterial incident detection using fixed detector and probe vehicle data. *Transportation Research Part C: Emerging Technologies*, 3(2):99-112, April 1995. URL <http://www.sciencedirect.com/science/article/B6VGJ-3YK014K-N/2/3954a40657e761121019db1ea5dfe9dd>.
- J.P. Singh, N. Bambos, B. Srinivasan, and D. Clawin. Wireless LAN performance under varied stress conditions in vehicular traffic scenarios. In *Proceedings, IEEE 56<sup>th</sup> Vehicular Technology Conference, Fall 2002, Volume 2*, pages 743–747. IEEE, 2002.
- Michael A. P. Taylor, Jeremy E. Woolley, and Rocco Zito. Integration of the global positioning system and geographical information systems for traffic congestion studies. *Transportation Research Part C: Emerging Technologies*, 8(1-6):257-285, 2000. URL <http://www.sciencedirect.com/science/article/B6VGJ-417F93H-H/2/bb1e78d7aad0bf3a0351b726ca16d4db>.
- N. E. Thomas. Multi-state and multi-sensor incident detection systems for arterial streets. *Transportation Research Part C: Emerging Technologies*, 6(5-6):337-357, December 1998. URL <http://www.sciencedirect.com/science/article/B6VGJ-3WM4P0T-4/2/a5f362640e99e24446c4d732f1f32ae0>.
- Hao Wu, Richard Fujimoto, Randall Guensler, and Michael Hunter. MDDV: a mobility-centric data dissemination algorithm for vehicular networks. In: *Proceedings of the first ACM Workshop on Vehicular Ad Hoc Networks*, pages 47 - 56, New York, NY, October 2004. ACM Press. URL <http://doi.acm.org/10.1145/1023875.1023884>.
- Athanasios K. Ziliaskopoulos and Jiang Zhang. A zero public infrastructure vehicle based traffic information system. In *82<sup>nd</sup> Annual Meeting of the Transportation Research Board, Washington, D. C., January 2003*. TRB.